Yandamuri Anantha Satya* et al.
(IJITR) INTERNATIONAL JOURNAL OF INNOVATIVE TECHNOLOGY AND RESEARCH
Volume No.4, Issue No.5, August – September 2016, 3585 – 3589.

VLSI Implementation of Area and Power Optimized Quantum Dot Cellular Automata Comparator

YANDAMURI ANANTHA SATYA
M.Tech Student, Dept Of ECE
KIET, Kakinada, East Godavari, A.P, India

PARABATHULA SATYA VANI
Asst.Professor, Dept Of ECE
KIET, Kakinada, East Godavari, A.P, India

Abstract: **Quantum-dot cellular automata (QCA) are a conspicuous technology suitable for the development of ultra-dense-low-power high-performance digital circuits. Efficient solutions have recently been proposed for several arithmetic circuits, such as adders, multipliers, and comparators. In this paper area and power optimized QCA comparator is presented for developing a 32bit full comparator. It is able to achieve lower area and power consumption. With respect to existing counterparts the comparators proposed here exhibit significantly higher speed and reduced overall area and power. The structures proposed in provide higher computational capabilities, and circuits able to separately recognize all the three possible conditions i.e., a = b, a > b, and a < b. The new strategy has been exploited in the design of two different comparator architectures and for several operands word lengths. The proposed scheme, we deal with 32-bit numbers with less number of resources unlike conventional comparators, which leads to the realization of low power and area efficient comparator.**

Keywords: **Quantum-Dot Cellular Automata; QCA Established Comparator; Novel QCA Comparator**

I. INTRODUCTION

Quantum dot Cellular Automata (QCA) technology provides a promising opportunity to overcome the approaching limits of conventional CMOS technology. For this reason, in recent years the design of logic circuits based on QCA has received a great deal of attention, and special efforts have been directed towards arithmetic circuits, such as adders, multipliers , and comparators.

Even though comparators are key elements for a wide range of applications, QCA implementations existing in the literature are mainly provided for comparing two single bits. Only few examples of comparators able to process n-bit operands, with n > 2, are available,. The comparator described in simply computes the XNOR function to establish whether two input bits a and b match each other. The structures proposed in provide higher computational capabilities, and circuits able to separately recognize all the three possible conditions in which a = b, a > b, and a < b (here named full comparators) are described. The 1-bit implementation proposed and then improved, has been exploited, to design a parallel n-bit full comparator. An example of serial structures is provided, whereas the n-bit comparator described and can recognize only the case in which, A and B being the n-bit inputs, A ≥ B. Alternative QCA implementations of 1-bit full comparators were recently proposed. With respect to other QCA designs, the latter exhibit reduced delays, area occupancy and number of used cells.

This paper focuses on the design of efficient parallel QCA-based n-bit full comparators. The main contribution of this paper is the introduction of a novel design methodology that allows low computational time and very compact layouts to be achieved. In particular, original theorems and corollaries are stated and demonstrated that directly impact on the QCA realizations of some basic Boolean functions used within the comparator architectures.

The novel theorems were applied to achieve innovative QCA-based structures of n-bit full comparators that were laid out and simulated using the QCADesigner tool for n ranging between 2 and 32. As an example, one of the 32-bit comparators designed exploiting the proposed theory is implemented using less than 2800 cells within an overall area of about 2.66 μm2; moreover, it requires only 15 clock cycles to complete the operation.

The rest of the paper is organized as follows: a brief back-ground of the QCA design approach and existing QCA implementations of binary comparators , the new theorems and corollaries are then enunciated and demonstrated ,comparators designed exploiting the novel theorems are proposed in this paper that also presents comparison results with existing designs.

II. QCA ESTABLISHED COMPARATOR

There are several QCA designs of comparators in the literature. A 1-bit binary comparator receives two bits *a* and *b* as inputs and establishes whether they are equal, less than or greater than each other. These possible states are represented through three output signals, here named $A_{e\ q}B$, $A_{b\ ig}B$, $B_{b\ ig}A$, that are asserted, respectively, when $a=b$, $a>b$, and $a < b$. Full comparators are those that can separately identify all the above cases, whereas non-full comparators recognize just one or two of them. As an example, the comparator designed in and depicted in Fig. 5(a) can verify only whether $a=b$. Conversely, the circuits shown in Fig. 5.1(b) and (c), proposed, are full comparators. The latter also

exploits two 1-bit registers *D* to process *n*-bit operands serially from the least significant bit to the most significant one.

With the main objective of reducing the number of wire crossings, which is still a big challenge of QCA designs , in the universal logic gate (ULG) $f(y_1, y_2, y_3) = M(M(y_1, y_2, 0), M(y_1, y_3, 1), 1)$was proposed and then used to implement the comparator illustrated in Fig. 1(d). It is worth noting that, two *n*-bit numbers $A_{(n-1:0)}=a_{n-1}...a_0...b_0$can be processed by cascading *n* instances of the 1-bit comparator. Each instance receives as inputs the *i*th bits a_i and b_i (with $i=n-1, ...,0$) of the operands and the signals $A_{big}B_{(i-1:0)}$ and $B_{big}A_{(i-1:0)}$. The former is asserted when the sub word $A_{(i-1:0)}=a_{i-1}...a_0$ represents a binary number greater than $B_{(i-1:0)}=b_{i-1}...b_0$. In a similar way ^{,B}big $^{A}(i-1:0)$ is set to 1 when $^{A}(i-1:0) < ^{B}(i-1:0)$. The outputs $A_{big}B_{(i:0)}$ and $B_{big}A_{(i:0)}$ directly feed the next stage. It can be seen that this circuit does not identify the case in which *A* = *B*, therefore it cannot be classified as a full-comparator.

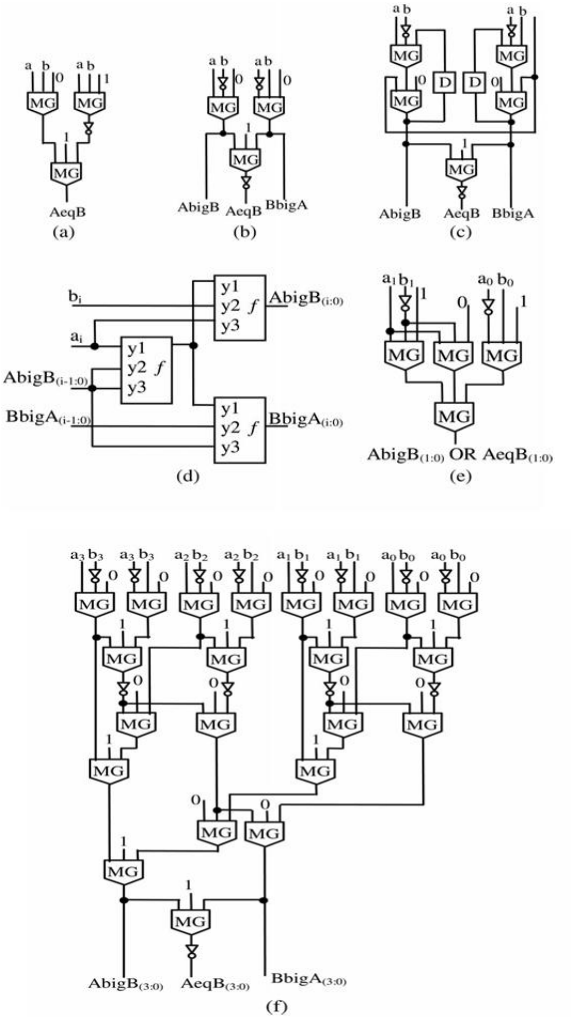

Fig 2.1 QCA established comparator presented in:(a),(b),(c),(d), (e), (f) .

The design described in exploits a tree-based (TB) architecture and exhibits a delay that in theory logarithmically increases with *n*. The 2-bit version of such designed comparator is illustrated in Fig. 5.1(e) also the full comparator proposed in exploits TB architecture to achieve high speed. As shown in Fig. 5.1(f), where 4-bit operands are assumed, one instance of the 1-bit comparator presented it is used for each bit position. The intermediate results obtained in this way are then further processed through a proper number of cascaded 2-input OR and AND gates implemented by means of MGs having one input permanently set to 1 and 0, respectively. Analyzing existing QCA implementations of binary comparators it can be observed that they were designed directly mapping the basic Boolean functions consolidated for the CMOS logic designs to MGs and inverters, or ULGs. Unfortunately, in this way the computational capability offered by each MG could be underutilized. As a consequence, both the complexity and the overall delay of the resulting QCA designs could be increased in vain.

2.1 NOVEL QCA COMPARATORS:

The first proposed comparator exploits a cascade-based (CB) architecture. To explain better how the overall computation is performed, the schematic diagram illustrated in Fig. 3 is provided. It shows a possible implementation of a 32-bit comparator based on the proposed theory. Following the criterion illustrated in Fig. 3, an *n*-bit CB full comparator designed as proposed here uses: *n*/3 instances of T1 and/or T2; *n*/3 cascaded instances ofT4 through which the signals $A_{bigB}(n-1:0)$ and $B_{bigA}(n-1:0)$ are computed; and one instance of C2, needed to compute also $A_{eqB}(n-1:0)$. Circles visible in Fig. 5.2 indicate the additional clock phases that have to be inserted on wires to guarantee the correct synchronization of the overall design. The CB full comparator was designed for operands word lengths ranging from 2 to 32 and using, for n > 2, the split criterion summarized in Table I. Obviously, alternative splits could be used.

As it is well known, the number of cascaded MGs within the worst computational path of a QCA design directly affects the delay achieved. In fact, each MG introduces one clock phase in the overall delay. From Fig.2.3 , it can be seen that the modules T1 and T2 contribute to the computational path with one inverter and two MGs. Each instance of T4 introduces one more MG, whereas C2 is responsible for one MG and one inverter. As a consequence, the critical computational path of the novel n-bit CB full comparator consists of n/3+ 3 MGs and 2 inverters. As an example, the 32-bit implementation depicted in Fig. 3 has the worst-case path made up of 13 MGs and 2 inverters.

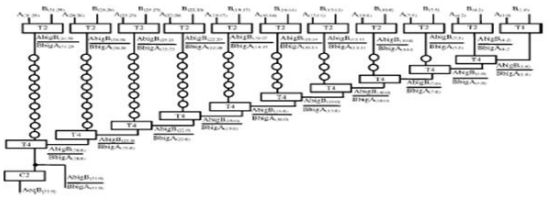

Fig 2.2 Novel 32-bit CB full comparator

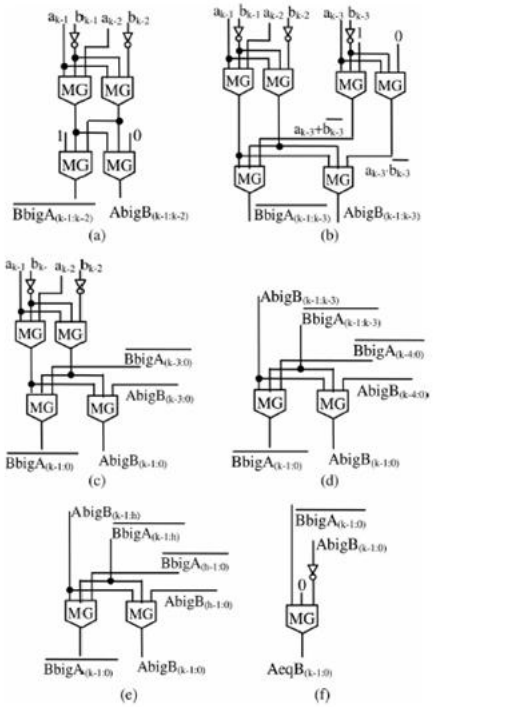

Fig. 2.3.QCA modules: (a) T1; (b) T2; (c) T3; (d) T4; (e) C1; and (f) C2

As always happens in CB computational architectures, the number of MGs within the computational path of the above-described comparator linearly increases with n. An alternative solution presented here adopts a TB architecture to achieve shorter computational paths. When this approach is exploited, several implementations of an n-bit full comparator can be de-signed differently combining the novel theorems and corollaries, as well as their QCA implementations depicted in Fig. 2. The TB comparators implement the comparison function recur-sively.

The operands A and B are preliminarily partitioned as $A = A_{msb}\ A_{lsb}$ and $B = B_{msb}\ B_{lsb}$. The portions A_{msb} and B_{msb} are compared independently of the portions A_{lsb} and B_{msb}. The depth of the recursion directly impacts the whole architecture. Examples of TB structures designed for 16- and 32-bit comparators are illustrated in Fig..2 In Fig. 2,3(b) and (d), the recursion with its minimum depth is adopted. The portions A_{msb} and B_{msb} , as well as the portions A_{lsb} and B_{lsb} , are separately compared trough two independent CB architectures. The overall result is finally built with the modules C1 and C2. Fig. 2.3(a) and (c) shows comparators designed adopting deeper recursions.

In the following of the paper, the 16- and 32-bit TB implementations illustrated in Fig.2.4(b) and (d) are deeply analyzed. Referring to the QCA modules depicted in Fig. 2, it can be easily verified that the former uses 35 MGs and 17 inverters and its critical computational path consists of 7MGs and 2 inverters, whereas the latter utilizes 83 MGs and 33 inverters and it has a worst-case path composed by 9 MGs and 2 inverters.

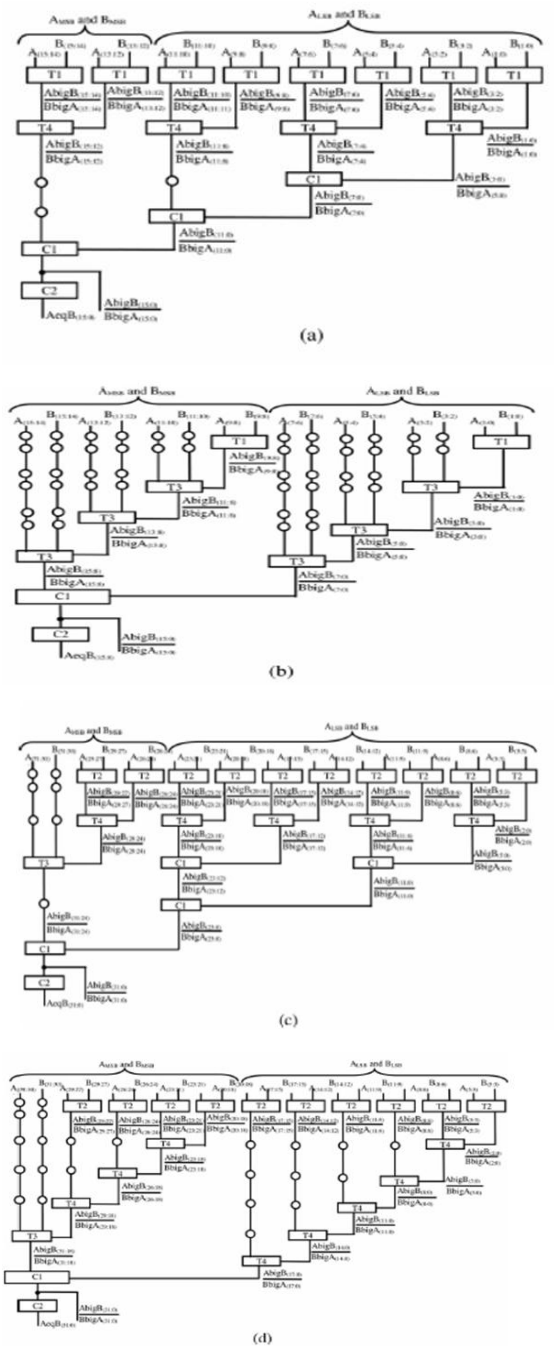

Fig. 2.4 Examples of novel TB comparators with: (a) and (b) 16-bit operands; (c) and (d) 32-bit inputs.

III. PROPOSED METHOD

The first proposed architecture presented in based on parallel approach and has two output bits H(A>*B*), S(i.e. *A*<*B*). The circuit for the 4-bit comparator is displayed in Fig.3.1 and is slenderly a modified version of the traditional comparator (which works on bit-weight comparison of two numbers from LSB to MSB) to understand the logic for the proposed architecture, let us consider an example for the comparison of $A=1011_2$ and $B=1100_2$.In the first stage, we identify and extract the 1s of first number which have a 0 in the corresponding position of the second number and are allowed to remain. The basic idea behind this is that only such 1s of a number make it greater than the other number. All other bit positions which have a1 in the corresponding position of the other number, are made 0.This is done for both the numbers in parallel ,that is ,*A* with respect to *B*(i.e. $A_i\overline{B_i}$) and *B* with respect to A(i.e. Bi, Ai') , there by forming two numbers *A'* and *B'* as shown

A = 1 0 1 1 *B* = 1 1 0 0 *A'*= 0 0 1 1 *B'*= 0 1 0 0

In the second stage, only the most significant 1s of *A'* and *B'*are extracted by giving it higher priority. Other 1s are made0. This stage incorporates logic similar to the *priority* logic of a priority encoder. This way two new numbers, *A''* and *B''*are formed as shown below. Due to the *priority* logic in corporate, the number of 1s in *A''* and *B''* is either one or zero.

A'=0 011	*B'*=0 100
A''=0 010	*B''*=0 10 0

In the last stage, from *A''* and *B''* two new signals are extracted. These are H (i.e. *A*>*B*) and S (i.e. *A*<*B*), both are of single bit, obtained by extracting the most significant bit (1) from *A''* and *B''*. If the 1 of *A''* is in a most significant position than that of *B''* or if *B''* has all 0s but *A''* has a 1, then this 1 is used to form output bit H. Similarly, if the 1 of *B''* is in a more significant position than that of *A''* or if *A''* has all 0s but *B''* has a 1, then this 1 is used to form output bit S as follows

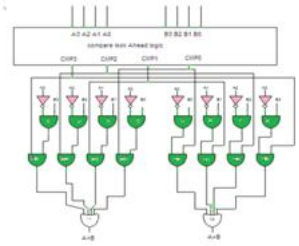

Fig. 3.1: Proposed diagram.

A''= 0 0 1 0	*B''*= 0 1 0 0
B''= 0 1 0 0	*A''*= 0 0 1 0
H = 0	S = 1

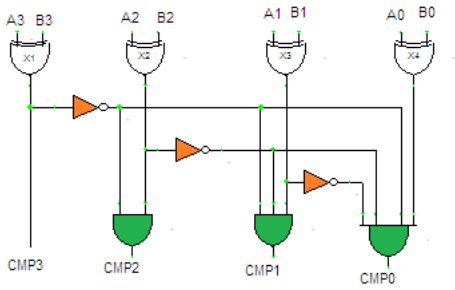

Fig. 3.2 Compare Look Ahead Logic

The schematic for 32-bit level implementation of the traditional and proposed comparators is shown in Figure 3.3. The blocks of the first stage compute the comparison result for every 4 bits of the input numbers. The blocks in the second stage take the result of four sets of 4-bit numbers and compute the result for the two 16-bit numbers which are obtained when the four sets of 4-bit numbers are concatenated. This logic is repeated in the third stage where the 2-bit block takes the results of two sets of 16-bit numbers and computes the result for the two 32-bit numbers.

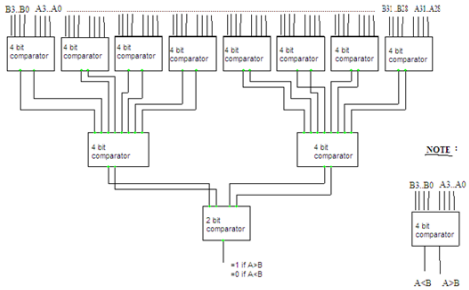

Fig.3.3:32-bit tree structure comparator

In the32-bit level implementation of both the proposed comparators, a modified 2-bit comparator module has been utilized. Since the numbers input to the 2-bit comparator module are the outputs of 4-bitcomparators, certain pairs of numbers can never be the input combinations: (10,10),(10,11),(11,10),(11,01),(01,11),(01,01).This is because the (A>B) and (A<B) output bits of the 4-bit comparator module can never be1 at the same time.

IV. RESULTS

4.1 Synthesis report:

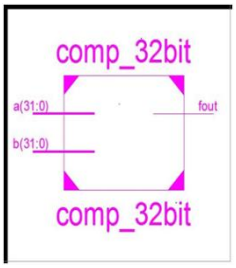

Fig 4.1 Synthesis report of 32 bit comparator

4.2 RTL Schematic:

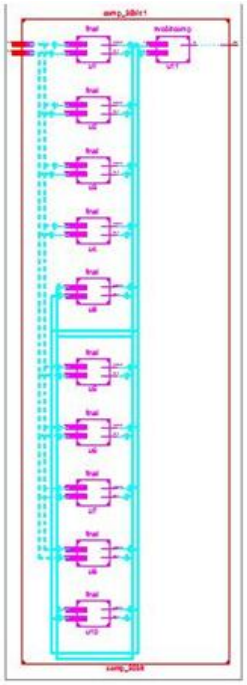

Fig 4.2 RTL Schematic

4.3 Technology Schematic:

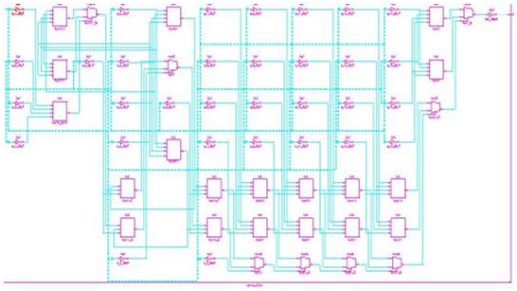

Fig 4.3 Technology Schematic

4.4 Resulted Waveform:

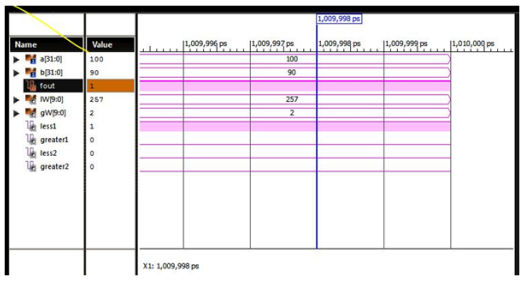

Fig 4.4 Output waveform

	No. of 4 input LUT's used	Delay (ns)	Power(mw)
EXISTING	61	16.091ns	0.4978mw
PROPOSED	19	14.419ns	0.1551mw

Fig4.5 comparing table

V. CONCLUSION

A digital comparator is an electronic device that takes two numbers as input in binary form and determines whether one number is greater than, less than or equal to the other number. In this proposed comparators have been presented, simulated and compared with the traditional one. Simulation results show maximum reduction in area, power and delay. We can conclude that proposed architecture for designing of the comparators are very efficient and be used efficiently. Design of area, power and delay forms the largest areas of a particular instance of a VLSI system design. Area and power can be reduced by using the above proposed design. We can use this comparator for the applications Arithmetic and Logic Unit(ALU),Digital signal processing etc.

VI. REFERENCES

[1] C. S. Lent, P. D. Tougaw, W. Porod, and G. H. Bernstein, "Quantum cellular automata," *Nanotechnology*, vol. 4, no. 1, pp. 49–57, 1993.

[2] M. T. Niemer and P. M. Kogge, "Problems in designing with QCAs: Layout= timing," *Int. J. Circuit Theory Appl.*, vol. 29, pp. 49–62, 2001.

[3] G. H. Bernstein,A. Imre,V.Metlushko,A.Orlov, L. Zhou, L. Ji,G. Csaba,and W. Porod, "Magnetic QCA systems," *Microelectron. J.*, vol. 36,pp. 619–624, 2005.

[4] J.Huang and F. Lombardi, *Design and Test of DigitalCircuits by Quantum-Dot Cellular Automata*. Norwood, MA, USA: Artech House, 2007.

[5] W. Liu, L. Lu, M. O'Neill, and E. E. Swartzlander Jr., "Design rules forquantum-dot cellular automata," in *Proc. IEEE Int. Symp. Circuits Syst.(ISCAS)*, Rio De Janeiro, Brazil, May 2011, pp. 2361–2364.

[6] K. Kim, K. Wu, and R. Karri, "Towards designing robust QCA architecturesin the presence of sneak noise paths," in *Proc. IEEE Design,Automation Test Eur. Conf. Exhib. (DATE)*,Munich, Germany,Mar. 2005,pp. 1214–1219.

[7] K. Navi, M. H. Moaiyeri, R. F. Mirzaee, O. Hashemipour, andB. M. Nezhad, "Two new low-power full adders based on majority-notgates," *Microelectron. J.*, vol. 40, pp. 126–130, 2009.

[8] H. Cho and E. E. Swartzlander Jr., "Adder design and analyses forquantum-dot cellular automata," *IEEE Trans. Nanotechnol.*, vol. 6, no. 3,pp. 374–383, May 2007. (QCA)

